|  |
| --- |
| FAST National University |
| **Subroutines**  **Lab 6** |
|  |

**Computer Organization and Assembly Language**

|  |  |
| --- | --- |
| **Student Name** | Eisha Amir |
| **Registration #** | 21L-5255 |
| **Instructor** | Hazoor Ahmad |
| **Class** | CS3 |
| **Section** | H |
| **Semester** | Fall 2022 |

Fast School of Computing

FAST-NU, Lahore, Pakistan

# Activity 1

## **Assembly Language Code**

[org 0x0100]

jmp start

data: dw 5,2,5,5

result dw 0

p dw 0

m dw 0

sum:

mov ax, [data+si]

add bx,ax

add si, 2

cmp si, 8

jne sum

mov[result], bx

shift:

shr bx, 1

shr bx,1

mov [p], ax

sub word[p], 1

mov word[m],ax

add word[m],3

ret

start:

mov si,0

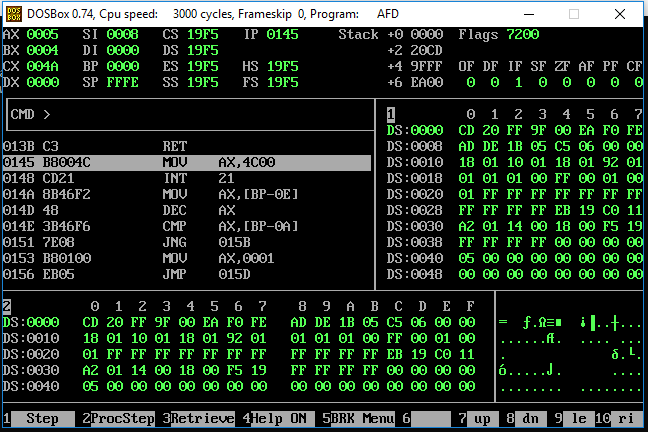
mov bx,0

call sum

mov ax, 0x4c00

int 21h

## **Debugging Screenshots**



# Activity 2

## **Assembly Language Code**

[org 0x0100]

jmp start

data :dw 0x5255

loop3: shr bx,loop2

cmp bp,3

jz loop4

sumshift: mov bp,4

loop1 : shr ax,1

rcr bx,1

dec bp

cmp bp,0

jz loop3

jne loop1

ret

start: mov si,4

mov ax,0x5255

loop2: mov bx,0

call sumshift

loop4: add di,bx

dec si

cmp si,0

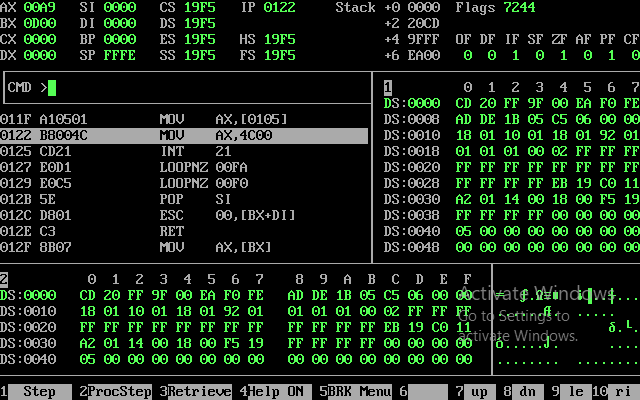
jne loop2

shr di,2

mov ax,0x4c00

int 21h

## **Debugging Screenshots**



# Activity 3

## **Assembly Language Code**

[org 0x0100]

jmp start

data: dw 5,2,5,5

result dw 0

p: dw 0

r: dw 0

res: dw 0

temp: dw 0

sum:

mov ax, [data+si]

add bx,ax

add si, 2

cmp si, 8

jne sum

mov[result], bx

shift:

shr bx, 1

shr bx,1

mov [p], ax

mov word[r],ax

sub word[r],2

ret

start:

mov si,0

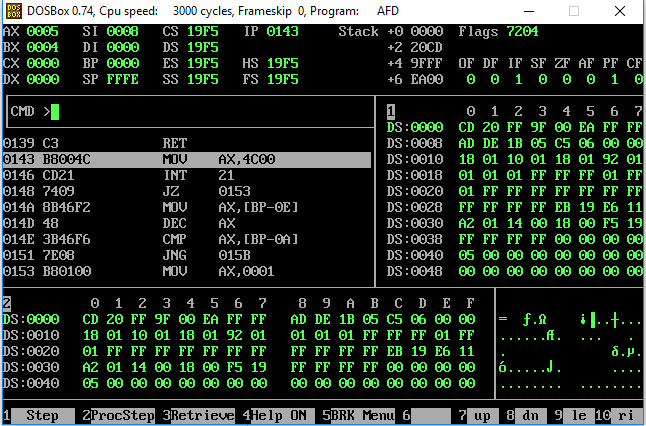
mov bx,0

call sum

mov ax, 0x4c00

int 21h

## **Debugging Screenshots**



# Activity 4

## **Assembly Language Code**

[org 0x0100]

jmp start

data: dw 5,2,5,5

result dw 0

p: dw 0

r: dw 0

res: dw 0

temp: dw 0

sum:

mov ax, [data+si]

add bx,ax

add si, 2

cmp si, 8

jne sum

mov[result], bx

shift:

shr bx, 1

shr bx,1

mov [p], ax

mov word[r],ax

sub word[r],2

factorial:

mov bx,[p]

mov dx, bx

mov cx, bx

dec cx

mov ax, 0

loop1:

add ax,dx

dec cx

jnz loop1

mov dx,ax

mov ax, 0

dec bx

mov cx, bx

dec cx

jnz loop1

mov[result], dx

ret

start:

mov si,0

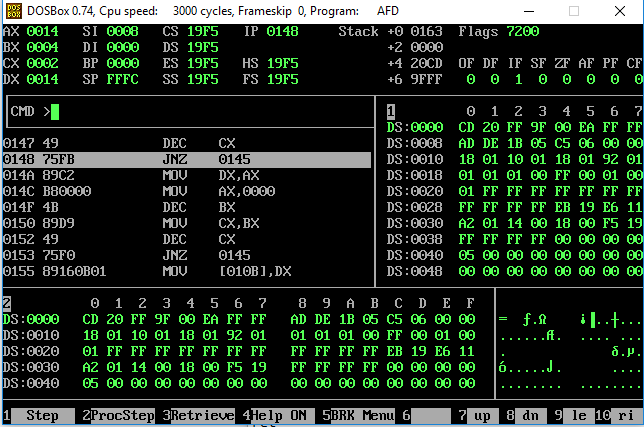
mov bx,0

call sum

mov ax, 0x4c00

int 21h

## **Debugging Screenshots**



# Activity 5

## **Assembly Language Code**

[org 0x0100]

jmp start

data: dw 5,2,5,5

result dw 0

p: dw 0

r: dw 0

res: dw 0

temp: dw 0

sum:

mov ax, [data+si]

add bx,ax

add si, 2

cmp si, 8

jne sum

mov[result], bx

shift:

shr bx, 1

shr bx,1

mov [p], ax

mov word[r],ax

sub word[r],2

factorial:

mov bx,[p]

mov dx, bx

mov cx, bx

dec cx

mov ax, 0

loop1:

add ax,dx

dec cx

jnz loop1

mov dx,ax

mov ax, 0

dec bx

mov cx, bx

dec cx

jnz loop1

mov[result], dx

factorial2:

mov bx, [p]

sub bx,[r]

mov dx, bx

mov cx, bx

dec cx

mov ax, 0

loop2:

add ax, bx

dec cx

jnz loop1

mov dx, ax

mov ax, 0

dec bx

mov cx, bx

dec cx

jnz loop2

mov word[temp], dx

ret

start:

mov si,0

mov bx,0

call sum

mov ax, 0x4c00

int 21h

## **Debugging Screenshots**

